

# Detecting Dominant Motion Flows and People Counting in High Density Crowds

Sultan Daud Khan, Giuseppe Vizzari, Stefania Bandini

Complex System and Artificial Intelligence,  
University of Milano-Bicocca

name.surname@disco.unimib.it

Saleh Basalamah

GIS Technology Innovation Center,  
Umm Al Qura University

smbasalamah@uqu.edu.sa

## ABSTRACT

Urbanisation is growingly generating crowding situations which generate potential issues for planning and public safety. This paper proposes new techniques of crowd analysis and precisely crowd flow segmentation and crowd counting framework for estimating the number of people in each flow segment. We use two foreground masks, one generated by Horn-Schunck optical flow, used by crowd flow segmentation, and another by Gaussian background subtraction, used by crowd counting framework. For crowd flow segmentation, we adopt  $K$ -means clustering algorithm which segments the crowd in different flows. After clustering, some small blobs can appear which are removed by blob absorption method. After blob absorption, crowd flow is segmented into different dominant flows. Finally, we estimate the number of people in each flow segment by using blob analysis and blob size optimization methods. Our experimental results demonstrate the effectiveness of the proposed method comparing to other state-of-the-art approaches and our proposed crowd counting framework estimates the number of people with about 90% accuracy.

## Keywords

Crowd analysis, clustering, crowd counting, crowd flow segmentation, crowd counting

## 1 INTRODUCTION

As the population of world is increasing and ever more located in urban areas, public safety is becoming a problem in most crowded areas of the big cities. Mass events like those related to sports, festivals, concerts, and carnivals attract thousands of people in constrained environments, therefore adequate safety measures must be adopted. Despite all safety measures, crowd disasters still occur frequently. The reasons of these disasters is mostly the presence of different and conflicting motion patterns that influence the crowd. A crowd is composed of small groups of people, for instance due to social relationships (families or friends) or a common goals, like reaching a certain point of the environment. The latter groups can be called short term coherent groups because they discontinue their cohesion after completing the goals (e.g. reaching an exit, completing a movement). Detecting the second kind of group, essentially associated to a certain flow of pedestrians in

the environment, can be important to be able to prevent conflict situations.

Due to the complex dynamics of the crowd, crowd management is becoming a daunting job where huge effort from the security staff is required to manage the potentially problematic situations. In such high density crowded areas, surveillance cameras are generally installed in different locations that can even cover the whole scene. Detecting specific activities in real-time videos is the task of analysts sitting in surveillance room and watching over multiple Tv screens. Such manual analysis of high density crowds is a tedious job and usually prone to errors. Therefore we need automatic analysis of the crowd which can reliably estimate the density and detect specific activities. Creating such kind of virtual analyst has become the focus of many researchers. This research has a wide range of application domain in crowd management, public space design, underwater fishes analysis (and animal behavior studies in general), and cell population analysis. In video surveillance, “detection and tracking” are the core technologies but these technologies are likely to fail in high density crowded scenarios. In this paper, we propose a framework that tackles problems of crowd flow segmentation, crowd counting and consists of three parts: foreground extraction, crowd flow segmentation, and crowd counting. The paper is organized as follows: the following section will briefly introduce related works. In Sect. 3 we shall describe propose framework. In

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Sect. 4 we shall discuss experimental results. Conclusions and future development will end the paper.

## 2 RELATED WORKS

For more than 40 years researchers have been studying pedestrian dynamics with the aim of supporting the design of pedestrian facilities. Since the population of the world is increasing and concentrating in urban areas, and due to the growing relevance of mass events (like sports contests, concerts and festivals periodically arranged and attracting growing number of people from different parts of the world), adequate safety measures are becoming ever more important. More recently, researchers are focusing on studying crowd dynamics in order to improve the evacuation strategies in emergency situations.

### 2.1 Motion Flow Segmentation

An important contribution that automated analysis tools can give to pedestrians and crowd safety is the detection of conflicting large pedestrian flows: this kind of movement pattern, in fact, may lead to dangerous situations and potential threats to pedestrians' safety. Therefore, segmenting typical flow patterns of crowd and estimating the number of people in crowd are important steps to understand overall crowd dynamics. Crowd flow segmentation has multiple benefits: (1) enables clutter free visualization of moving groups; (2) it is independent from "detection and tracking"; (3) provides input for the pedestrian simulation models (in terms of data for simulation initialization or validation). Automatic analysis of the crowd has become the center of focus for most of researchers in computer vision. Detecting pedestrians and tracking are traditional ways of crowd analysis. Most algorithms developed for object detection and tracking work well with pedestrians in low density crowds where the number of people is generally less than twenty individuals in a single frame, but with higher densities (where the number of people in a frame can be in the order of hundreds), detection and tracking of individuals are almost impossible due to multiple occlusions.

Therefore, the research has focused on gathering *global motion information* at higher scale. Global analysis of dense group of moving people is often based on *optical flow analysis*. [AS07] proposed particle dynamic segmentation of crowd flows by detecting the lagrangian coherent structures over the phase space. But their proposed method is computationally expensive because of the calculation of FTLE and also could not detect small flows. [OYA10] used SIFT features to detect dominant motion flows. Flow vectors of SIFT features are calculated and then motion flow map is divided into small regions of equal size. In each region, dominant motion flows are estimated by clustering flow vectors.

[EB08] proposed spectral clustering method for crowd flow segmentation by computing sparse optical flow field. Crowd flow is estimated using multiple visual features reported by [SND11] where flow is estimated by the number of persons passing through a virtual trip wire and accumulate the total number of foreground pixels. Min-cut/max flow algorithm is used by Ullah et al. [UC12] for crowd flow segmentation. In all of above four methods, we can not find clear boundaries among different flows. Crowd flow segmentation by using histogram curves is reported by [LRZ12] where angle matrix of foreground pixels is segmented instead of optical flow foreground. The derivative curve of histogram is used to segment the flow. Since this method only looks to the peaks of histogram curve, therefore it loses information about the crowd flow. Our proposed approach for detecting dominant flows is similar to spectral clustering in [EB08]. The difference is that we carry out segmentation by employing  $K$ -means clustering on dense optical flow field. After  $K$ -means clustering, small blobs can appear especially at the boundaries of conflicting flows, generated by the optical flow computation but not really associated to actual pedestrian movements; these small blobs are removed by our blob absorption method. Comparing to other approaches, our approach can detect large as well as small flows, and by employing blob absorption method, we generate clearer boundaries between different flow segments. We show the effectiveness of our proposed motion segmentation approach by comparing with state-of-the-art approaches in Sect. 4.

### 2.2 Crowd Counting

Most of the literature in field of crowd density estimation has focused on segmentation of *people or head counts*. Some of the work focused on texture analysis or wavelet descriptor for estimating crowd density. Bayesian based segmentation proposed by [ZN03] to segment and count the people but this method fails in high density scenarios as because of severe occlusions. [YST10] extract blob features of moving objects and neural network is trained to estimate number of pedestrians in each blob. [XLH06] classified crowd density into four classes by using wavelet descriptors. [MHL08] used texture descriptors to estimate crowd density. [TYOY99] count the number of people as they cross some virtual line. [HMY<sup>+</sup>97] used infra-red imaging to count the number of people in crowd. Simple background subtraction from static images to estimate the crowd density was proposed by [RP07]. Background removal concept is used to estimate the crowd area by [VYD<sup>+</sup>93, VYD<sup>+</sup>94]. [RMAS04] used a forward facing camera mounted on the car to detect crowd of pedestrians. Trained support vector machines using HAAR transform is used by [LCC01] to identify heads of people. Median background computing technique

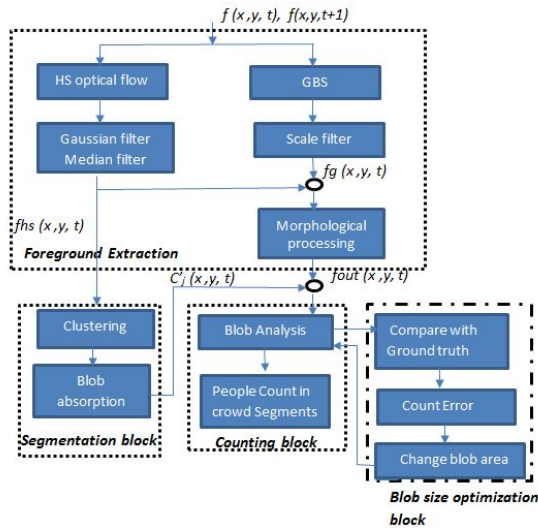


Figure 1: Overview of proposed framework

is used by [RP07] to extract foreground pixels. Support vector machine, K-nearest neighbor, PNN, BPNN are used to classify images in two categories (zero persons, one or more persons). As the sensor are becoming cheap, therefore, recently many researchers count the people using infra-red sensors. [TS07] proposed lightweight camera sensor nodes to count the people in the indoor environment based on motion histogram. Recently many infra-red sensors specifically designed for people counting are available in the market<sup>1</sup>. Our approach starts from the results of motion segmentation to perform people counting, estimating the number of pedestrians in each dominant flow by blob analysis and blob size optimization methods. This allows having a more selective and informative information on where the counted pedestrians are headed. Moreover, compared to the above methods, our approach is aimed at supporting people estimation even in high density crowds. As we will show in Sect. 4 we achieve satisfactory performance on people counting in experiments performed adopting different videos.

### 3 PROPOSED FRAMEWORK

Our proposed framework is composed of four processing blocks, Foreground extraction, segmentation, counting and blob size optimization block, but this block only executes in the beginning for few initial frames. In this section, we will discuss each processing block in detail. For sake of description of the proposed approach we will employ videos taken from a crowd related data set from UCF [AS07].

<sup>1</sup> See e.g. <http://www.sensourceinc.com/thermal-video-imaging-people-counters.htm> or <http://www.irisys.co.uk/people-counting/our-products/>.

### 3.1 Foreground Extraction

Foreground extraction is the most important pre-processing step for detecting the moving objects from the video and therefore forms the basis of our framework. Foreground extraction is useful for detection, tracking and understanding the behavior of the object. A survey on motion detection techniques can be found in [MG01]. Traditionally, in video surveillance with a fixed camera, researchers use background subtraction method, where foreground objects are extracted from video if the pixels in the current frame deviate significantly from the background. In this paper, we use two foreground masks as in [LWMZ10], one generated by optical flow,  $f_{hs(x,y,t)}$  and will be used by crowd flow segmentation framework and other is Gaussian background subtraction,  $f_{g(x,y,t)}$  used by counting framework as shown in Figure 1. Two consecutive frames  $f_{(x,y,t)}$  and  $f_{(x,y,t+1)}$  are applied to foreground extraction block. First, we compute Horn and Schunk (HS from now on) optical flow between adjacent frames, then Median filter and Gaussian filter are used to remove noises. We then set a threshold to get foreground mask  $f_{hs(x,y,t)}$ . In the same way, Gaussian Background Subtraction (GBS from now on) is used to get another foreground mask  $f_{g(x,y,t)}$ , after applying scale filter. Usually crowded objects move in wide areas, and for crowd flow segmentation, we need to detect change in every pixel, so optical flow methods reported in literature to compute sparse optical flow using the interest points (Lucas-Kanade optical flow) [LK<sup>+</sup>81] or dense optical flow for all pixels (HS optical flow) [HS81] in each frame can be used. Since, we want to detect change in every pixel, we compute dense optical flow. Since the optical flow vector of each pixel has the magnitude and direction values, we use magnitude information to extract foreground, all the pixels which have higher magnitude than  $T_{th}$  will be classified as foreground. Direction information of optical flow vectors can be used in crowd flow segmentation by clustering all optical flow vectors having similar orientations. Such methods are usually prone to errors due to unpredictable behavior of the pixels which change due to fast/slow moving objects and illumination. A small change in illumination can be detected as foreground objects even in the static background. Such methods can be useful in extracting region of interest (ROI) in the scene but can not be used in separating individuals in high density scenarios. As shown in Figure 2,  $f_{hs(x,y,t)}$  can not provide information about the group of foreground pixels (blobs) related to the people in the crowd. Therefore, for counting framework, we generate another foreground mask  $f_{g(x,y,t)}$  by Gaussian background subtraction method. GBS is a kind of background subtraction method [SG99] and is very good in separating objects from the background. GBS method is effective in suppressing

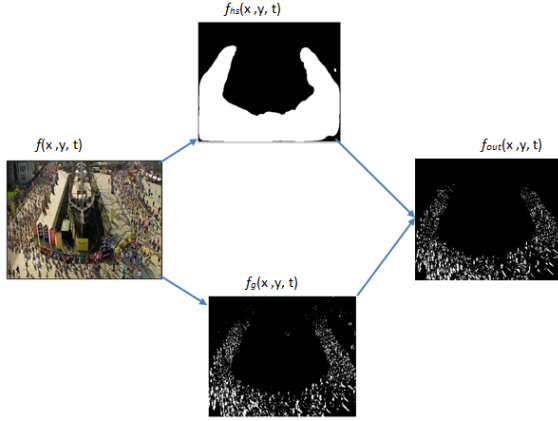


Figure 2: Foreground extraction framework

noise and robust to change in illumination.  $f_g(x, y, t)$  is also a binary image, where blobs represents the objects of different sizes. Small blobs are related to parts of object, medium blobs related to objects and large blobs represent group of objects, appeared due to occlusions. Optimal foreground mask  $f_{out}(x, y, t)$  is obtained by logical product of  $f_g(x, y, t)$  and  $f_{hs}(x, y, t)$ . Later on, we apply morphological processes like morphological opening and closing on the binary image  $f_{out}(x, y, t)$ . The morphological open operation is erosion followed by dilation, eliminates smooth contours and protrusions. While morphological close is dilation followed by erosion, smooths the section of contours, eliminates small holes and fills gaps in contours. These operations are dual to each other. Segmentation block segments the crowd flows into different clusters,  $C'_{j(x, y, t)}$ , by employing  $K$ -means clustering followed by blob absorption method. To estimate the number of people in each flow segment, we take logical product of each cluster  $C'_{j(x, y, t)}$  and foreground mask  $f_{out}(x, y, t)$  and count the number of people by blob analysis and blob size optimization methods.

### 3.2 Motion Flow Field Computation

After foreground extraction, the objects in the foreground move in different directions as shown in first row of Figure 3. It can be seen that in each video, foreground objects have multiple flows. Since we use dense HS optical flow that computes movement of every pixel, we call it motion flow field. The motion flow field is a set of independent flow vectors in each frame and each flow vector is associated with its respective spatial location. This instantaneous motion field of the video contains temporal information and can be used for the learning motion pattern of the video. Consider a feature point  $i$  in  $F_t$ , its flow vector  $Z_i$  includes its location  $X_i = (x_i, y_i)$  and its velocity vector  $V_i = (v_{x_i}, v_{y_i})$ , i. e.  $Z_i = (X_i, V_i)$  where  $\theta_i$  is the angle or direction of  $V_i$ , where  $0^\circ \leq \theta \leq 360^\circ$ . Then  $\{Z_1, Z_2, \dots, Z_k\}$  is the motion flow field of all the foreground points of an image.

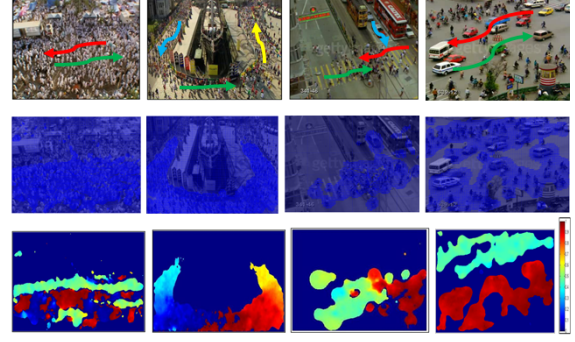


Figure 3: First Row: sample frames from videos of the Hajj, a marathon, pedestrian crossing, and road section; second row: corresponding optical flow; third row: corresponding direction map

#### 3.2.1 Motion Flow Field Segmentation

The motion flow field  $\{Z_1, Z_2, \dots, Z_n\}$  is a  $n \times 4$  matrix where each row represents flow vector  $i$  and columns represents its spatial location  $X_i$  and velocity vector  $V_i$ .  $n$  represents total number of flow vectors (foreground points). Each flow vector represents motion in specific direction as shown in Figure 3, third row. Figure 3, (third row) does not show dominant motion patterns, so we can not infer any meaningful information about flows. Therefore, we need a method that automatically analyses the similarity among the flow vectors and cluster them in multiple groups. We use  $K$ -means clustering algorithm (widely used in data analysis and image segmentation) to segment motion flow field into different groups. This process of grouping vectors that represent specific motion pattern is called segmentation. After segmentation process, motion field is divided into multiple segments. We denote  $K$  as the initial number of cluster centroids. Commonly used initialization methods are Forgy and Random Partition [HE02]. We initialize cluster centroids as  $(K - 1) \times 360^\circ / K$ . let  $C = \{1, 2, \dots, j\}$  is the set of initial cluster centroids.  $\epsilon = 360^\circ / 2K$  and  $\delta = 360^\circ / K$ .

##### Step 1 Clustering with initial K-centroids

```

for  $1 \leq i \leq n$  do
  for  $1 \leq j \leq K$  do
    if  $\|\theta_i - c_j\| \leq \epsilon$  then, where  $c_j \in C$ 
       $z_i(x_i, v_i) \rightarrow c_j$ 
       $n_j \leftarrow n_j + 1$ 
    end if
  end for
end for

```

##### Step 2 New centroids calculation

```

for  $1 \leq j \leq K$  do
   $c'_j = \sum_{i=1}^{n_j} \theta_i / n_j$ , Update  $C$  with new centroids  $c'_j$ 
end for

```

##### Step 3 Clustering of similar clusters

```

if  $\|c'_l - c'_m\| \leq \delta$  then

```





Figure 4: Results of 4-means clustering in a Hajj video frame

$$c'_l = \sum_{i=1}^{n_l+n_m} \theta_i / n_l + n_m$$

$$c'_m \leftarrow c'_l$$

end if

**Step 4** Return to step 1

This approach can be applied to the images where the objects moves in every direction. For such kind of complex movements in images, we assign larger value of  $K$  while we assign lower value to the images where objects move in regular directions. In this paper, we assign lower value of  $K = 4$  because in our benchmark videos, objects move in regular directions. Figure 4, shows that the objects in sample frame are clustered into different groups by applying 4-means clustering. We use different colors to differentiate clusters. Let  $C = \{1, 2, \dots, K\}$  is the set of clusters found in sample frame.

### 3.2.2 Blob Absorption

We noticed that after  $K$ -means clustering, some small blobs appear: these small blobs represent small clusters as shown in Figure 4 and resulted due to following reasons. First, if the objects move slowly, the inside and outside flow vectors of the objects are not same and as a result are classified into two different flows. Second, if the two opposite optical flow intersect, the optical flow at the boundaries is ambiguous. Third, small blobs represents small groups of people and are not the part of dominant motion flows and they are not relevant to the aims of our analysis. Therefore, we adopt blob absorption approach (mimicking a “big fish eats small fish” process), where these blobs are either absorbed by dominant cluster or by the background. The algorithm is as follows:

1. Compute weights for all clusters, i.e.  $C_{wi} = \sum_{j=1}^K n_j / T$ . where  $n_j$  is number of features points  $z(x, v)$  in cluster  $C_j$  and  $T$  is total number of foreground points.
2. Select cluster  $C_j$  and perform blob analysis and find area of each blob in  $C_j$ .
3. Use threshold area  $L$  and find blobs whose area  $A \leq L$ . Let  $B = \{b_1, b_2, \dots, b_n\}$  set of blobs represents small clusters and needs to be absorbed.
4. Select blob  $b_i$  from set  $B$ , find its edges points by using canny edge detector [Can86].

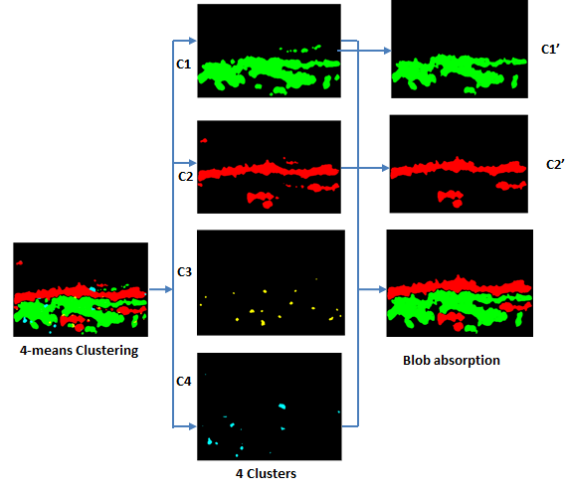


Figure 5: Results of the Blob Absorption method applied to a frame of the Hajj video

5. For each edge point, look at its neighborhood points, find neighborhood cluster ids and store ids of neighborhood points in array  $S$ . Remove those points from  $S$  that have same cluster id  $j$ , because  $b_i$  can not be absorbed by itself.
6. From remaining points in  $S$ , compute blob weight  $b_{wi} = \sum_{j=1}^N n_j / T_s$ . where  $N$  is the total number of cluster ids found in  $S$ .  $n_j$  is total number of points with cluster id  $j$  and  $T_s$  is total number of points in  $S$ .
7. Compute  $w_i = c_{wi} + b_{wi}$  and cluster id  $j$  with maximum weight  $w_i$  is selected and id  $j$  is assigned to all points of blob  $b_j$ . Hence blob is absorbed.
8. Repeat steps 4 to 7 until  $B$  is empty.
9. Repeat step 2. Here background is also considered as cluster with id and cluster weight  $c_w = 0$ .

After blob absorption, as shown in Figure 5, small clusters ( $C_3$  and  $C_4$ ) are removed leaving behind large clusters ( $C_1$  and  $C_2$ ) representing dominant flows with clear boundaries, by setting up threshold area  $L = 500$ . Let  $C' = \{1, 2, \dots, j\}$  is set of large clusters.

### 3.3 Counting People in High Density Crowds

This section describes the methodology for counting people in high density crowds. In this step, we count the number of people in each cluster  $C'_j$ . In low density crowds, due to clear visibility of individual with little occlusions, we can detect, track and count the number of individuals in crowd, but in high density crowds, it is hard to extract and count the individuals due to (i) with increasing density, the number of pixels/individual

decreases (ii) severe occlusions result in the loss of observation of the target individual (iii) discerning individuals from one another is caused by constant interaction among individuals in a crowd. Therefore, as a solution, we perform blob analysis and blob size optimization techniques on foreground image and estimate the number of people in high density crowds.

### 3.3.1 Blob Analysis and Blob Size Optimization

For extracting foreground, belonging to each dominant flow (or cluster  $C'_j$ ), we take logical conjunction of each cluster  $C'_j$  and foreground mask  $f_{g(x,y,t)}$ , generated by Gaussian background subtraction and shown in Figure 7. First row of Figure 7, shows that sample frame of marathon video is segmented into three dominant flows while second row shows foreground elements belonging to each of three segments. After foreground extraction, small blobs appear which represent moving objects. Blobs are the connected regions of variables “areas” in the binary image. Since there are many blobs of different areas representing different moving objects we need to find an optimal area that will serve as a threshold. The blob with areas above this threshold will not be taken into account (for instance, when counting pedestrians in road videos, these large blobs might be related to cars). For computing threshold area we devised blob size optimization algorithm discussed below.

1. Select the blob's size randomly. let's blob's size is  $A$ .
2.  $c_i = \text{blobAnalysis}(A)$ ; will return count of blobs whose size  $\leq A$  for frame  $i$ .
3.  $\text{error}_j = \|c_i - \text{gth}_i\|$ . where  $\text{gth}_i$  is the ground truth count for frame  $i$ .
4. Vary the blob size  $A$  by some constant  $k$  and repeat step 2 to 4 for  $N$  iterations.
5. Select blob's size  $A$  for which  $\text{error}_j$  is minimum.

Note that for finding optimum blob size, we used only four or five initial frames whose ground truth is available. These frames are selected randomly. For each initial frame we compute optimum blob size by using the method discussed above. We take the mean  $A'$  of all four or five optimum sizes computed for each initial frame and use  $A'$  for counting people in rest of frames. Average and standard deviation of the error between people count using the blob area and the actual number of people (Ground Truth) is plotted in Figure 6 versus blob area. In Figure 6, mean and standard deviation of the counting error is plotted for a road video. It can be seen from the figure that the error is minimum for the blob area 17, resulting therefore context dependent. It must be stressed that the optimal blob size depends on the video, especially on the point of vantage determining the size in pixels of people to be counted (in other

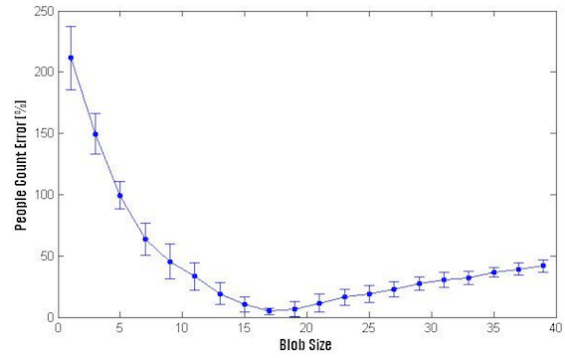


Figure 6: Blob size optimization for Road video: notice that the optimal blob size for error minimization is different for different videos.

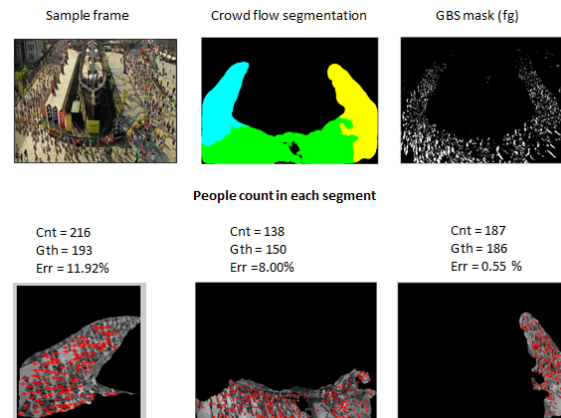


Figure 7: People Counting Framework highlighting results of intermediate steps in one frame of the marathon video

videos analysed in Sect. 4 the optimal blob size is as small as 2 pixels). Through experiments, we observed that for small blob areas, the count of people will be higher as the noise will also be counted as people. For large blob areas, instead, some people might be missed in the count. Hence selection of optimal blob size is very important to minimize the error in people count.

## 4 EXPERIMENTAL RESULTS

This section presents the quantitative analysis of the results obtained from experiments. We carried out our experiments on a PC of 2.6 GHz (Core i5) with 4.0 GB memory and data set from UCF [AS07]. The data set covers two types of crowded scenarios: the first scenario consists of videos involving high density crowds i.e. videos from Hajj and a marathon, where the number of people is higher than 150 in a single frame. The second scenario covers low density crowds where the number of people in a frame is lower than 70, i.e. road crossing video, where people are moving over zebra crossing in different directions, and road video, where vehicles and people are moving in different directions on road. Since our framework consists of two major

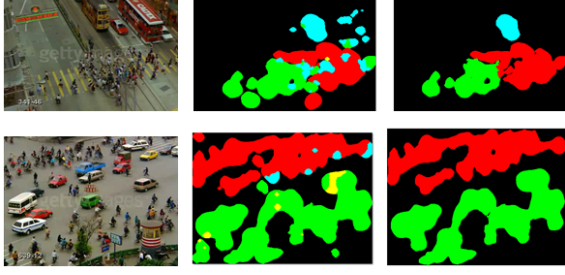


Figure 8: First column: sample frames; Second Column: K-means clustering results; Third Column: Blob absorption results

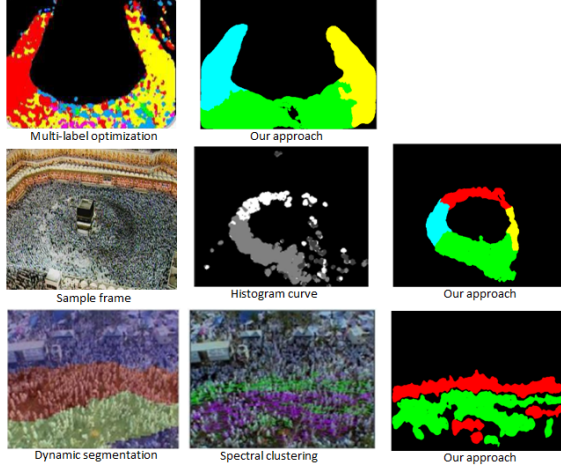


Figure 9: Comparing Results

parts, crowd flow segmentation and crowd counting, our experiments are carried out in two steps.

#### 4.1 Segmentation Results

We selected 65 frames from each video. After computing optical flow, we apply K-means clustering algorithm that cluster all the similar flow vectors. In this paper, we use  $K = 4$  for all the videos, so after segmentation, we detect four different flows in video frame as shown in second column of Figure 8. We then apply blob absorption method to remove small clusters as shown in third column of Figure 8. For blob absorption we use different threshold  $L$  values. Small clusters can not be absorbed completely by using smaller values of  $L$  while we lost some portions of dominant cluster by using larger values of  $L$ . Therefore, we determined value of  $L$  experimentally and is different for different videos. After blob absorption, image of cross video is segmented into three flows, red (west), green (east) and cyan (south). While image of road video is segmented into two flows, red and green as shown in third column of Figure 8.

We compared our approach in Figure 9 with multi-label optimization [UC12], histogram curve [LRZ12], dynamic segmentation [AS07] and spectral clustering [EB08]. In the first row of Figure 9, we compare our

Table 1: Hajj Video people counting in sequence of frames

F.n.	G.T.(E)	G.T.(W)	Cnt.(E)	Cnt.(W)	Err.(E)	Err.(W)
12	151	159	170	154	12,58%	3,14%
20	153	161	167	154	9,15%	4,35%
29	185	185	195	194	5,41%	4,86%
37	176	187	192	201	9,09%	7,49%
45	187	186	200	191	6,95%	2,69%
55	187	187	195	188	4,28%	0,53%
63	189	185	194	194	2,65%	4,86%
Average Error					7,16%	3,99%

Table 2: Crossing video people counting in sequence of frames

F.n.	G.T.(E)	G.T.(W)	Cnt.(E)	Cnt.(W)	Err.(E)	Err.(W)
10	30	30	30	29	0,00%	3,33%
16	34	35	30	39	11,76%	11,43%
22	37	36	25	38	32,43%	5,56%
28	35	33	29	32	17,14%	3,03%
30	38	35	37	43	2,63%	22,86%
35	38	34	35	41	7,89%	20,59%
40	37	36	36	39	2,70%	8,33%
47	35	36	35	30	0,00%	16,67%
55	37	38	38	34	2,70%	10,53%
64	37	40	31	28	16,22%	30,00%
Average Error					9,35%	13,23%

Table 3: Road Video people counting in sequence of frames

F.n.	G.T.(E)	G.T.(W)	Cnt.(E)	Cnt.(W)	Err.(E)	Err.(W)
11	45	67	33	44	26,67%	34,33%
20	38	65	45	58	18,42%	10,77%
30	42	62	46	69	9,52%	11,29%
35	41	61	40	62	2,44%	1,64%
43	39	64	36	53	7,69%	17,19%
50	40	65	48	67	20,00%	3,08%
55	40	65	36	55	10,00%	15,38%
62	39	63	39	67	0,00%	6,35%
Average Error					11,84%	12,50%

method with multi-label optimization method. We see that crowd flow segmentation using multi-label optimization could not segment the crowd into dominant flows. Moreover, it could not find clear boundary due to small blobs appeared after segmentation. In the second row of Figure 9, we compare our results with histogram curve method. Segmentation by using histogram curve is fastest than existing methods but it lost much information about the crowd flows, since this method only looks to the peaks of histogram curves. In the third row of Figure 9, we compare our results with dynamic segmentation and spectral clustering approach. Dynamic segmentation is not able to detect small flows in the crowd, while spectral clustering carries out segmentation on sparse optical flow and give the approximate segmentation where we can not find clear boundaries between flows. All the above shortcomings are resolved by our proposed approach. Our proposed approach not only detects dominant flows but can also detects small flows without the loss of crowd flow information. Moreover, our proposed approach finds clear boundaries among different flows.

#### 4.2 Crowd Counting Results

After crowd flow segmentation, we count the number of people in each flow segment. Each video consists

Table 4: Marathon Video people counting in sequence of frames

F.n.	G.T.(E)	G.T.(N)	G.T.(S)	Cnt.(E)	Cnt.(N)	Cnt.(S)	Err(E)	Err(N)	Err(S)
11	145	192	187	134	176	199	7,59%	8,33%	6,42%
15	150	186	193	138	187	216	8,00%	0,54%	11,92%
20	148	193	200	126	178	190	14,86%	7,77%	5,00%
27	155	200	211	151	244	225	2,58%	22,00%	6,64%
33	150	195	220	145	223	219	3,33%	14,36%	0,45%
39	160	205	210	151	199	222	5,63%	2,93%	5,71%
45	158	210	205	145	215	210	8,23%	2,38%	2,44%
49	156	207	210	145	189	197	7,05%	8,70%	6,19%
55	162	215	215	164	210	196	1,23%	2,33%	8,84%
59	158	220	220	162	210	202	2,53%	4,55%	8,18%
62	167	225	224	158	185	198	5,39%	17,78%	11,61%
Average Error							6,04%	8,33%	6,67

of sequence of 65 frames and our proposed method automatically counts the number of people in each frame as shown in Tables 1, 2, 3, 4. Tables show counting results of random frames taken from each analysed video, where F.n. represents frame number of the analysed sequence. The rise and fall in people count in different frames represents the fact that people are entering or leaving the scene affecting people count at different time. To check the counting accuracy of the proposed framework, ground truth (G.T) for each direction (East(E), West(W), North(N), South(S)) is found for the frames after random intervals and count error (Err) is computed by comparing results with the ground truth data. Count error is shown in details in tables 1, 4, 2, 3 for all analyzed video sequences. The first column of each table shows the frame number, G.T. shows ground truth found for each direction and Cnt. is counting results of our proposed approach. Average error is less than 12% for all analyzed video sequences. For some frames, however, count error is higher due to the fact that some people in that frame are missed in count or noise (resulted after motion segmentation) is counted as people. As obvious from tables, our proposed framework works better in high density scenarios like Hajj and marathon. It is matter of the fact, that in high density scenarios, people covers much of the scene's area in comparison to low density scenarios. After motion segmentation, foreground extracted in high density scenarios contains less background noise (foreground noise generally moves with people and it is not causing significant errors) in comparison to foreground extracted in low density scenarios. From the experimental results, it is clear that our proposed approach count the people in each video sequence with 90% accuracy.

To study the time complexity of our proposed framework, we utilize 65 frames of each of four analysed videos and time is recorded as average frame processing time and recorded in Table 5. The latter shows time complexity of crowd flow segmentation and crowd counting frameworks. Rows of table shows the analysed videos and column represents time complexity of each of processing block. It is obvious that clustering takes much time as compare to blob absorption method and crowd counting framework. It is matter of the fact

that  $K$  means clustering is computationally expensive and can be very slow to converge in worst case scenarios, i.e. high resolution videos, and high ratio of foreground to background pixels. In this paper, we use videos of the same resolution, 360x480. Although the resolution of all analysed videos is same, yet time complexity is different. The ratio of foreground to background pixels of different videos is different and usually the ratio is higher if the large part of the scene is covered by foreground pixels. It is also obvious from table that Hajj video takes more computational time than other videos. It is matter of the fact that most of scene of a Hajj video frame is covered by foreground pixels than background pixels. The computational time can be reduced and proposed framework can be employed in real time, if implemented in openCV. The current implementation is in Matlab.

## 5 CONCLUSIONS

In this paper, we have considered both high and low density crowds and proposed a framework that automatically detects dominant motion flows and counts the number of people in each flow. Such kind of analysis provides a useful input to pedestrian simulation models. A first employment of the our analysis is related to the actual initial configuration of the simulation scenario. Second way to exploit data resulting from automated video analysis is represented by pedestrian counting and density estimation: the indication of the average number of pedestrians present in the simulated portion of the environment is important in configuring the start areas. Finally, we can use the above analysis in the validation of simulation results. Our approach is applicable in many different situations and it is independent of local conditions and camera viewpoints. Our method does not require detection and tracking of people, hence preserving the privacy of the people. Future works are aimed at improving the precision, especially in low density situations, but also aim at using these techniques to more comprehensively characterize the movement patterns in the analyzed frame by identifying and quantitatively describe points of entrance of pedestrians, points of interest and exits, essentially to derive a so called origin-destination matrix.



Table 5: Time Complexity of our proposed framework in (seconds)

Videos	Crowd Flow Segmentation		Crowd Counting		
	Clustering	Blob Absorption	Seg # 1	Seg # 2	Seg # 3
Marathon	6	2.77	0.006	0.007	0.005
Hajj	9.88	2.93	0.009	0.008	NIL
Road	7.02	1.67	0.005	0.004	NIL
Crossing	5.12	1.03	0.003	0.005	NIL

## 6 REFERENCES

- [AS07] Saad Ali and Mubarak Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *CVPR*. IEEE Computer Society, 2007.
- [Can86] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.
- [EB08] Günther Eibl and N Brandle. Evaluation of clustering methods for finding dominant optical flow fields in crowded scenes. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.
- [HE02] Greg Hamerly and Charles Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 600–607. ACM, 2002.
- [HMY<sup>+</sup>97] Kazuhiko Hashimoto, Katsuya Morinaka, Nobuyuki Yoshiike, Chjihiro Kawaguchi, and Satoshi Matsueda. People count system using multi-sensing application. In *Solid State Sensors and Actuators, 1997. TRANSDUCERS'97 Chicago., 1997 International Conference on*, volume 2, pages 1291–1294. IEEE, 1997.
- [HS81] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.
- [LCC01] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao. Estimation of number of people in crowded scenes using perspective transformation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 31(6):645–654, 2001.
- [LK<sup>+</sup>81] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.
- [LRZ12] Wei Li, Jiu-Hong Ruan, and Hua-An Zhao. Crowd movement segmentation using velocity field histogram curve. In *Wavelet Analysis and Pattern Recognition (ICWAPR), 2012 International Conference on*, pages 191–195. IEEE, 2012.
- [LWMZ10] Wei Li, Xiaojuan Wu, K Matsumoto, and Hua-An Zhao. Crowd foreground detection and density estimation based on moment. In *Wavelet Analysis and Pattern Recognition (ICWAPR), 2010 International Conference on*, pages 130–135. IEEE, 2010.
- [MG01] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.
- [MHL08] Wenhua Ma, Lei Huang, and Changping Liu. Advanced local binary pattern descriptors for crowd estimation. In *Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on*, volume 2, pages 958–962. IEEE, 2008.
- [OYA10] Ovgu Ozturk, Toshihiko Yamasaki, and Kiyoharu Aizawa. Detecting dominant motion flows in unstructured/structured crowd scenes. In *ICPR*, pages 3533–3536. IEEE, 2010.
- [RMAS04] Pini Reisman, Ofer Mano, Shai Avidan, and Amnon Shashua. Crowd detection in video sequences. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 66–71. IEEE, 2004.
- [RP07] Damian Roqueiro and Valery A Petrushin. Counting people using video cameras. *The International Journal of Parallel, Emergent and Distributed Systems*, 22(3):193–209, 2007.
- [SG99] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.
- [SND11] Satyam Srivastava, Ka Ki Ng, and Edward J. Delp. Crowd flow estimation using multiple visual features for scenes with changing crowd densities. In *AVSS*, pages 60–65. IEEE Computer Society, 2011.
- [TS07] Thiago Teixeira and Andreas Savvides. Lightweight people counting and localiz-

- ing in indoor spaces using camera sensor nodes. In *Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on*, pages 36–43. IEEE, 2007.
- [TYOY99] K Terada, D Yoshida, Shunichiro Oe, and J Yamaguchi. A method of counting the passing people by using the stereo images. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 2, pages 338–342. IEEE, 1999.
- [UC12] Habib Ullah and Nicola Conci. Crowd motion segmentation and anomaly detection via multi-label optimization. In *ICPR workshop on Pattern Recognition and Crowd Analysis*, 2012.
- [VYD<sup>+</sup>93] SA Velastin, JH Yin, AC Davies, MA Vicencio-Silva, RE Allsop, and A Penn. Analysis of crowd movements and densities in built-up environments using image processing. In *Image Processing for Transport Applications, IEE Colloquium on*, pages 8–1. IET, 1993.
- [VYD<sup>+</sup>94] SA Velastin, JH Yin, AC Davies, MA Vicencio-Silva, RE Allsop, and A Penn. Automated measurement of crowd density and motion using image processing. In *Road Traffic Monitoring and Control, 1994., Seventh International Conference on*, pages 127–132. IET, 1994.
- [XLH06] Li Xiaohua, Shen Lansun, and Li Huanqin. Estimation of crowd density based on wavelet and support vector machine. *Transactions of the Institute of Measurement and Control*, 28(3):299–308, 2006.
- [YST10] Satoshi Yoshinaga, Atsushi Shimada, and Rin-ichiro Taniguchi. Real-time people counting using blob descriptor. *Procedia-Social and Behavioral Sciences*, 2(1):143–152, 2010.
- [ZN03] Tao Zhao and Ramakant Nevatia. Bayesian human segmentation in crowded situations. In *CVPR (2)*, pages 459–466. IEEE Computer Society, 2003.